

## GUEST EDITORIAL

# Computational design synthesis

MATTHEW I. CAMPBELL<sup>1</sup> AND KRISTINA SHEA<sup>2</sup>

<sup>1</sup>School of Mechanical, Industrial and Manufacturing Engineering, Oregon State University, Corvallis, Oregon, USA

<sup>2</sup>Engineering Design and Computing Laboratory, ETH Zurich, Zurich, Switzerland

Nearly every human society has been fundamentally altered by computers in the last 30 years. We rely on servers, desktops, and handheld devices to facilitate duties of both our personal and professional lives. Most of this is just *managing data*: input, store, transfer, output. There is also *analysis*: given a set of inputs, solve for  $x$ . For engineers,  $x$  is a prediction of how hot, how much stress, how much efficiency, or how much cost. However, other than the use of computers to manage data or perform analysis, there is a third use of computation that is rarely perceived by the average computer user or engineer, and that is the use of computers in *design synthesis*. As our engineering artifacts grow in complexity, we need to offload some design decisions to the computer. We need the computer to help us synthesize many of the minute details in our engineering devices as well as ensure high performance by searching among a myriad of alternatives for the optimal combination of building blocks and parameter values.

Computational design synthesis (CDS) is a research area focused on approaches to automating synthesis activities in design. Resulting methods may be fully automated or interactive with the goals of automatically generating a range of alternatives, sparking creativity and innovation, automating tedious or time-consuming engineering tasks, and simply exploring the creative abilities of computational systems. There is a fuzzy line between CDS and applied optimization. We have intended CDS to be more ambitious than the typical use of optimization to “solve for  $x$ .” It is intended to mimic what humans consider in design, not only parameters, like in a fixed vector,  $x$ , but also material choices, discrete component choices, and the basic architecture of building blocks.

Such research is typically ambitious in scope, demanding in terms of developmental and computational resources, and extensive in terms of related work. The work is based on artificial intelligence, mathematical programming, computational geometry, graph theory, engineering design theory,

and cognitive science. When applied later in the design process, meaningful results are only achievable by interfacing with the computational analysis tools that govern our engineering world, such as those for solving partial-differential field equations (e.g., finite element analysis or computational fluid dynamics) or those for solving ordinary-differential equations (e.g., three-dimensional dynamics). In addition to the technical challenge, it can be difficult to interface with these tools because many are expensive, deal with proprietary file formats, and are sometimes operable only within the tool’s graphical user interface. Combine this with highly iterative optimization methods (many design synthesis techniques require the use of large stochastic optimization methods), and researchers may occasionally find themselves up against practical limits in computational time and memory.

Early in the design process, CDS has faced issues of how to represent and reason with nebulous notions of function and feasibility as well as methods of predicting performance when various parameters are undefined. In the first paper, “Evaluating FuncSION: A Software for Automated Synthesis of Design Solutions for Stimulating Ideation During Mechanical Conceptual Design,” Ujjwal Pal, Ying Chieh Liu, and Amaresh Chakrabarti address this problem directly by synthesizing transformations (without the need for detailed part definition) to the problem of converting energy from one distinct location to another. It is a classic example of a body of work that has grown steadily over the last three decades. It is our opinion, and likely the opinion of the paper’s authors, that this work must someday positively affect the design of electromechanical systems.

CDS has also championed the use of generative design grammars as a means to simultaneously provide structure and design freedom during synthesis. A generative grammar is composed of rules that, unlike the traditional definition of rules, focus on defining the actions or design transformations and modifications that can be performed. Just like human languages (e.g., English, German, or Chinese), formal grammars require a vocabulary of terms that can take the form of strings, parameters, graphs, or shapes, which can also be represented by graphs. Unlike an expert system, the grammar rules are more about capturing design logic concisely than about

Reprint requests to: Matthew I. Campbell, School of Mechanical, Industrial and Manufacturing Engineering, 408 Rogers Hall, Oregon State University, Corvallis, OR 97331-6001, USA. E-mail: [matt.campbell@oregonstate.edu](mailto:matt.campbell@oregonstate.edu)

detailed expertise and define a search space of valid designs. The definition of rules is yet to be a straightforward or simple process, and two papers in this Special Issue delve into the grammar development and application process. In the second paper, “Systematic Rule Analysis of Generative Design Grammars,” Corinna Königseder and Kristina Shea investigate how to support the rule development process and show how the choices made in rule development have a significant effect on the quality and complexity of the defined search space. The third paper, “Layout Synthesis of Fluid Channels Using Generative Graph Grammars,” shows how the definition of grammar rules provides an alternative to competing methods that optimize thousands of variables simultaneously. Although the editors are also authors on these two papers, the combination of generative grammars and design synthesis appears to be an auspicious and exciting approach in yielding impressive results and defining powerful software in the future.

CDS is not limited to a single problem domain. The first three papers deal with mechanical design problems, and the latter three address related fields: computational strategy design, architectural design, and electronics design. The fourth paper, “Evolutionary Computational Synthesis of Self-Organizing Systems,” presents an opposing approach to the one adopted in the first three. Instead of rules dictating a space of solutions that are searched by a single decision maker, James Humann, Newsha Khani, and Yan Jin present a multi-agent (multidecision maker) approach. Here, agents perceive fields and make quick decisions based on improving an individual agent’s current state. Although the decisions may seem myopic, the distribution of decisions appears to have promise in terms of efficiency while still leading to quality solutions.

Architectural design has always been a close cousin to engineering design and manufacturing, because of similar issues, including the evaluation of multiple objectives and complex spatial relationships. In “Generative Spatial Performance Design System,” Benjamin P. Coorey and Julie R. Jupp summarize an architectural system to room layout that uses graphs to both synthesize and analyze solutions. One added challenge faced in architectural design is the multitude of non-quantifiable objectives. Coorey and Jupp point out the importance of *precedence* (similarity to previous solutions) as a way to judge the merit of a solution. It is a concept that engineers may want to consider in complex problem domains.

Finally, our Special Issue concludes with a comparative study in electronic packaging design. Because of the magnitude of code development and testing needed to produce a single opus (as in the earlier papers), there is too little effort put on comparing different search approaches to CDS (although our second paper, by Königseder and Shea, presents a thorough comparison of how representational differences

effect search). “Comparing Two Methods Based on Evolutionary Algorithms for Generating Layouts: Dense Packing Versus Subdivision” shows how two wildly different approaches yield similar results. The insights can be used to choose a method depending on the design scenario.

In reading the Special Issue, we hope that readers get a sense of the exciting challenges and opportunities that exist in this budding research area. As humans continue to design more complex systems, structures, products, and buildings, the reliance on CDS will likely increase. As a research community, we should continue to define, formalize, and categorize our approaches to effectively automatically and interactively design the complexities of our engineered and built environment.

---

**Matthew I. Campbell** is a Professor of mechanical engineering at Oregon State University. He received his PhD from Carnegie Mellon University in 2000 with honors and membership in Phi Kappa Phi and Pi Tau Sigma. For more than 15 years, he has focused on methods that independently create solutions for typical mechanical engineering design problems like gear trains, sheet metal, planar mechanisms, and planning for manufacturing, assembly, and disassembly. As such, he has become a world-class expert in a variety of fields, such as machine design, design theory, artificial intelligence, graph theory, and numerical optimization. Dr. Campbell has over 100 published articles and has been acknowledged with best paper awards at ASME, American Society for Engineering Education, and Design Society conferences. His research focuses on automating difficult or tedious engineering design tasks.

**Kristina Shea** is a Professor of engineering design and computing at ETH Zurich. She studied mechanical engineering at Carnegie Mellon University, where she completed her PhD in 1997. She has held appointments as a Postdoctoral Researcher at EPFL, Switzerland; a University Lecturer at University of Cambridge; and Associate Professor at Technische Universität München. Dr. Shea carries out internationally leading research in new computational models, methods, and tools for design with a focus on supporting early stage design, including synthesis, optimization, and fabrication, and has published over 100 papers with several best paper awards. Her expertise is more specifically in formal, integrated product models, graph and shape grammars, multiobjective and multidisciplinary optimization, and digital design to fabrication. Professor Shea is an Associate Editor for *AIEDAM* and the *ASME Journal of Mechanical Design*. She serves on the board of management of the Design Society and is a fellow of ASME.